

Time	Narration
00:01	به spoken tutorial در Scripts and Functions با Scilab خوش آمدید.
00:06	با مقدمه‌ای کوتاه در file formats در سایلِب شروع می‌کنیم.
00:12	وقتی که تعدادی commands باید اجرا شوند این راحت‌تر است که آن‌ها را یک فایل با ویرایشگر سایلِب بنویسیم.
00:21	به این‌ها فایل‌های SCRIPT گفته می‌شود.
00:24	برای اجرای این فرمان‌ها در فایل script می‌توان از عملگر exec که به همراه نام فایل script می‌باشد استفاده کرد.
00:34	این فایل‌ها معمولاً پسوند sce یا sci. بسته به محتویات آن را دارند.
00:42	فایل‌هایی که پسوند sce. را دارند شامل عملگرهای سایلِب و یا عملگرهای تعریف شده توسط کاربر می‌باشند.
00:51	با اجرای این فایل‌ها عملگرها را در محیط سایلِب قرار می‌گیرند. (اما اجرا نمی‌شوند) در صورتی که
01:00	فایل‌هایی که پسوند sce. را دارند شامل عملگرهای سایلِب و یا عملگرهای تعریف شده توسط کاربر می‌باشند.
01:08	به خاطر داشته باشید قرارداد نامگذاری با پسوندهای sce و sci. قانون نمی‌باشد فقط یک قراردادی است که توسط انجمن سایلِب دنبال می‌شود.
01:21	پنجره کنسول سایلِب را در کامپیوتر باز می‌کنیم.
01:27	دایرکتوری کاری فعلی را با تایپ کردن فرمان pwd در command prompt بررسی می‌کنیم.
01:35	به Task bar پنجره کنسول سایلِب بروید و گزینه Editor را برای بازکردن ویرایشگر (editor) سایلِب کلیک کنید.
01:49	من از قبل commands را در فایل تایپ کرده ام و به صورت helloworld.sce ذخیره کرده‌ام. از این به بعد این فایل را با استفاده از آیکون میانبر Open a file باز می‌کنم.
02:03	فایل helloworld.sce را انتخاب کنید و Open را کلیک کنید.
02:10	شما ممکن است که فرمان را در فایل جدید تایپ کنید و آن فایل را در دایرکتوری کاری فعلی به صورت helloworld.sce از طریق منوی File ذخیره کنید.
02:20	به دکمه Execute در نوار منو scilab editors بروید و گزینه Load into Scilab را انتخاب کنید.
02:29	این فایل را در کنسول سایلِب load می‌کند. (قرار می‌دهد)
02:34	بعد از قرار دادن فایل در کنسول script خروجی را به این صورت که می‌بینید ایجاد می‌کند.
02:43	این شامل فرمان‌ها و نتیجه خروجی برای command های موجود می‌باشد.
02:49	حالا مقدار a را به 1 تغییر دهید.
02:55	در editor به منوی File بروید و Save را کلیک کنید.
03:02	ما همچنین می‌توانیم script را مستقیماً از scilab interpreter با استفاده از دستور exec و مسیر داده شده به script file به این صورت اجرا کنیم:
03:12	helloworld.sce into double quotes exec into brackets که این نام فایل می‌باشد. Enter را فشار دهید.
03:31	فایل script خروجی مثل همان استفاده از عملگر exec را می‌دهد.
03:37	درمورد functions صحبت می‌کنیم.
03:39	تعریف function با کلیدواژه function شروع می‌شود و با کلیدواژه endfunction خاتمه می‌یابد.

03:46	من از قبل function file را در function.sci با استفاده از scilab editor ذخیره کرده ام.
03:57	من آن فایل را باز می کنم.
04:03	همانطور که می بینید function در اینجا تعریف شده است.
04:08	اینجا degrees پارامتر خروجی و radians پارامتر ورودی می باشد.
04:21	با نام function (عملگر) radians2degrees
04:26	من این function را در سایلِب با استفاده از گزینه Execute menu بارگذاری می کنم.
04:40	این تابع حالاً در کنسول سایلِب بارگذاری شده است.
04:44	این همچنین با استفاده از دستور exec می تواند بارگذاری شود.
04:47	یکبار که تابع بارگذاری شد، آن را می توان مانند یک تابع سایلِب با گذراندن arguments به آن تابع فراخوانی کرد.
04:56	یک یادداشت برداری ذهنی از علامت درصد (%) داشته باشید و دلیل استفاده از آن را به یاد بیاورید.
05:02	حالا مقادیر $\text{radians2degrees of } \%pi/2$ و $\text{radians2degrees of } (\%pi/4)$ را بدست می آوریم.
05:17	$\text{percent pi}/2$ و $\text{percent pi by } 4 (\%pi/4)$.
05:28	ما حالا تابع با بیش از یک ورودی و argument های خروجی را می بینیم.
05:33	این تابع polar coordinates را بعنوان argument ورودی می گیرد و rectangular coordinates را بعنوان arguments خروجی می دهد.
05:44	من فایلی را که از قبل تایپ کرده ام باز می کنم.
05:51	در اینجا می بینید که x و y پارامترهای خروجی هستند و r و theta پارامترهای ورودی برای تابع polar2rect می باشند.
06:06	من این تابع را با استفاده از گزینه exec در سایلِب بارگذاری می کنم.
06:21	یکبار که تابع بارگذاری شد ما باید تابع را فراخوانی کنیم. این تابع به دو arguments ورودی و دو arguments خروجی نیاز دارد.
06:31	بنابر این $r = 2$;
06:37	$\text{theta} = 45$;
06:44	و حالا آن را x1 comma y1 پارامترهای خروجی مساوی نام تابع polar2rect into bracket r comma theta فرا می خوانیم. Enter را فشار دهید.
07:25	شما مقادیر x1 و y1 را می بینید.
07:29	یکی از ویژگی های جالب سایلِب این است که می توانید هر عدد از توابع را در یک فایل .sci تعریف کنید.
07:38	با انجام این به یاد داشته باشید که به صورت پیش فرض همه متغیرها که در تابع (function) تعریف می شوند local (محلی) هستند و محدوده این متغیرها که در یک تابع خاص استفاده می شوند با کلیدواژه endfunction از تعریف تابع خاتمه می یابند.
07:55	فواید این ویژگی این است که ما می توانیم از یک نام متغیر در توابع مختلف استفاده کنیم.
08:05	این متغیرها با هم مخلوط نمی شوند مگر اینکه از گزینه سراسری (global) استفاده شود.
08:10	برای دانستن بیشتر در مورد متغیرهای سراسری help global را تایپ کنید.
08:18	توجه کنید که اگر متغیری قرار باشد در داخل تابع دیده و یا نظارت شود disp لازم می باشد.

08:26	داخل file function می‌توانید تأثیر قرار دادن semicolon (;) (نقطه ویرگول) در پایان statement را خودتان ببینید.
08:34	همچنین این را برای disp statements بررسی کنید.
08:38	Inline Functions: Functions بخشی از کد می‌باشد که شامل ورودی و خروجی به خوبی تعریف شده و متغیرهای داخلی می‌باشد.
08:46	ساده‌ترین روش برای تعریف function (تابع) استفاده از دستور deff. می‌باشد.
08:53	سایلب اجازه ایجاد in-line functions را می‌دهد که بخصوص وقتی که بدنه function کوتاه باشد مفید است.
09:02	این را می‌توان با کمک تابع deff() انجام داد.
09:07	این دو پارامتر رشته ای را می‌گیرد.
09:10	رشته اول رابط به function را تعریف می‌کند و رشته دوم بیانیه function را تعریف می‌کند.
09:19	دستور deff که function را در سایلب تعریف و همچنین بارگذاری می‌کند.
09:26	نیازی نیست که function را که با دستور deff از طریق گزینه execute menu تعریف شده است بارگذاری (load) کنیم.
09:34	با یک مثال این را توضیح می‌دهیم.
09:41	من فایل inline.sci را که در آن inline function را نوشته‌ام باز می‌کنم.
09:51	من پنجره editor را اندازه دوباره می‌دهم.
09:57	همانطور که قبلاً گفته شد رشته اول اعلام function را تعریف می‌کند و رشته دوم بیانیه function را تعریف می‌کند.
10:13	ما این function (تابع) را در Scilab editor بارگذاری (load) می‌کنیم تا مقادیر degrees2radians of 90 و degrees2radians of 45 را بدست آوریم.
10:54	Function نه فقط بقیه توابع (function) بلکه خودش (itself) را نیز باید فراخوانی کند.
11:00	این فراخوانی recursive از function (تابع) می‌باشد.
11:03	این لازم می‌باشد مثلاً وقتی که یک function (تابع) برای محاسبه فاکتوریل یک عدد صحیح می‌نویسیم.
11:10	این توضیحات را در فرمتهای فایل در سایلب گسترش می‌دهیم.
11:14	همانطور که گفته شد سایلب دو فرمت فایل با نامهای Sce. و فرمت فایل sci. را استفاده می‌کند.
11:23	فایلها با پسوند sce. که script files می‌باشند که شامل فرمانهای سایلب که شما در طول یک جلسه تعاملی سایلب وارد کرده‌اید می‌باشد.
11:35	آنها می‌توانند شامل خطوط فرمان استفاده شده در مستند سازی function باشند و همچنین می‌توانند از فرمان exec برای اجرای اسکریپت استفاده کنند.
11:52	فایل ها با پسوند فایل sci. فایل های تابع هستند که با دستور function شروع می شوند.
12:00	یک فایل sci. که می تواند تعاریف چندگانه ای برای function داشته باشد که خودشان شامل تعدادی Scilab statements هستند، که پس از ارزیابی، عملیات را بر روی آرگومان های function یا متغیرهای خروجی انجام می دهند.
12:20	این ما را به پایان برنامه آموزشی در Scripts and Functions در سایلب می آورد.

12:25	توابع زیاد دیگری در سایلب می‌باشد که در برنامه‌های دیگر توضیح می‌دهیم.
12:31	لینکهای سایلب را مشاهده کنید.
12:33	این spoken tutorial توسط Free and Open Source Software in Science and Engineering Education (FOSSEE) ایجاد شده است.
12:40	اطلاعات بیشتر در مورد پروژه FOSSEE را از http://fossee.in یا http://scilab.in بدست آورید
12:50	این تحت پشتیبانی National Mission on Education از طریق ICT, MHRD دولت هند می‌باشد.
12:56	اطلاعات بیشتر در http://spoken-tutorial.org/NMEICT-Intro موجود می‌باشد.
13:06	ترجمه و صداگذاری شب‌نم اقبال خان
13:10	با تشکر از شما